Integrated Distributed Authentication Protocol for Smart Grid Communications

Neetesh Saxena, *Member, IEEE,* and Bong Jun Choi, *Member, IEEE*

Abstract—In the smart grid, an integrated distributed authentication protocol is needed to not only securely manage the system but also efficiently authenticate many different entities for the smart grid communications. In addition, a lightweight authentication protocol is required to handle frequent authentications among billions of devices. Unfortunately, in the literature, there is no such integrated protocol that provides mutual authentications among the home environment, energy provider, gateways, and advanced metering infrastructure network. Therefore, in this paper, we propose a lightweight cloud trusted authorities-based integrated (centrally controlled) distributed authentication protocol that provides mutual authentications among various communicated entities in a distributed manner. Based on certificateless cryptography, our protocol is lightweight and efficient even if there are invalid requests contained a batch. Security and performance analysis show that the protocol provides privacy preservation, forward secrecy, semantic security, perfect key ambiguous, and protection against identity thefts while generating lower overheads in comparison with the existing protocols. Also, the protocol is secure against man-in-the-middle attacks, redirection attacks, impersonation attacks, and denial-of-service attacks. Moreover, our protocol provides a complete resistance against the flood-based denial-of-service attacks in the smart grid.

Index Terms—Authentication, Smart grid, DoS attacks, Redirection attacks, Cloud computing.

1 Introduction

The smart grid (*SG*) is a critical infrastructure, whose objective is to provide more efficient, secure, stable, and reliable power to consumers, operators, and utilities. The *SG* system for home environment consists of various components such as smart meters (*SM*), home appliances (*HA*), energy providers (*EP*), gateways (*GW*), and advanced metering infrastructure (*AMI*) network. It is generally assumed that home area network (*HAN*) is wirelessly connected with the Zigbee technology [1], whereas the building area network (*BAN*)/neighborhood area network (*NAN*) is connected by wide area network (*WAN*) technologies and cellular technologies, such as global system for mobile communication (*GSM*) and long term evolution (*LTE*) [2]. Smart meters can be considered to be equipped with two communication interfaces, where one interface works as a *SM* and the other works as a *HAN-GW*. Therefore, the *SM* is a central home controller that communicates with all the *HA* within a household. Further, *BAN-GW*/*NAN-GW* acts as (or deploy) an aggregator (*AG*) that receives data from the *SM* and forwards it to the respective control center (*CC*) via relays and concentrators with via wired/wireless connections. Figure 1 shows the overall architecture of the *SG* system.

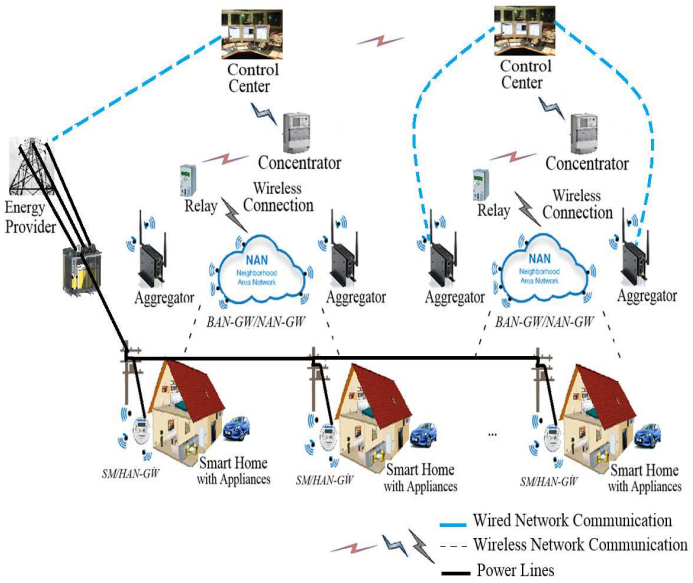


Fig. 1: Overall architecture of the smart grid system.

Two-way communications in the *SG* enable instant interaction between different *SG* entities and help to improve the overall efficiency of the *SG* system. However, without proper authentications, the system resources and entities can be compromised that may result in financial losses and performance degradation [3], [4]. Therefore, in the *SG* system, an integrated, distributed, fast, and lightweight authentication protocol is needed to provide mutual authentications between the various entities of the *SG* system. An

- *N. Saxena is with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, USA. E-mail: mr.neetesh.saxena@ieee.org.*
- *B.J. Choi is with the Department of Computer Science, State University of New York (SUNY) Korea, Incheon, South Korea and jointly with the Department of Computer Science, Stony Brook University, NY, USA. E-mail: bjchoi@sunykorea.ac.kr.*

This research was funded by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the "ICT Consilience Creative Program" (IITP-2015-R0346-15-1007) supervised by the IITP (Institute for Information & Communications Technology Promotion) and under the "Basic Science Research Program" (2013R1A1A1010489, 2015R1C1A1A01053788) through the NRF (National Research Foundation).
Manuscript received xxx xx, xxxx; revised xxx xx, xxxx.

integrated distributed protocol can maximize the utilization of shared resources while generating low overhead. In the *SG* system, a fast and lightweight protocol is required as the authentication process is frequently repeated many times among billions of devices. Further, the security protocol of the *SG* system must provide defense against the known security attacks, including man-in-the-middle (*MITM*) and denial-of-service (*DoS*) attacks [5]. According to the *NIST* reports, one of the main security issues in the *SG* system is that the authentication mechanism does not sufficiently authenticate devices or exposes authentication keys. Therefore, a centralized authentication in decentralized environment is required for the centralized security management in terms of event logging/analysis and authentication [6], [7].

There are many different communication protocols used for delivering various commands and control information. These protocols were not designed initially with security in mind. Today, when Internet is connected to the *SG* system, various organizations, such as *ETSI*, *IEEE*, and *NIST* are embedding security to the existing protocols as new standards in order to prevent the system against various well known security attacks. However, they need to modify many communication standards for making them security embedded. This creates additional overheads. Furthermore, researchers have not yet focused much on an integrated protocol, rather they have proposed separate protocols for individual connections between different entities in the *SG*. They have not discussed the integration of these protocols for compatible communication among them. This motivates us to propose such an efficient and secure authentication protocol for the *SG* system. However, there are several challenges in designing an integrated distributed authentication protocol as we identified below:

- The protocol should not only be controlled by a central entity, but also by the subsystems of the *SG* network in a distributed manner.
- Embedding security solutions in each communication protocol of the *SG* network not only are highly complex but also generate huge overhead and cost. Therefore, it would be more flexible and efficient to instead design a cyber-security layer over the communication network to maintain end-to-end security [8]. This simplifies the integration at the cyber-security layer, rather than integrating different communication protocols in the *SG* network.
- The protocol should be able to fully utilize the available system resources.
- The generated overhead by the protocol should be as low as possible. The protocol should be fast and lightweight, as the authentication process is frequently repeated many times among billions of devices, especially, when devices receive multiple messages at once, such as when gateways/aggregators authenticate multiple smart meters and gather data from them.
- The protocol must utilize the suitable cryptosystem (with symmetric and/or asymmetric keys) as recommended by various standard organizations, such as *IEEE*, *ETSI*, and *NIST*. Particularly, *NIST* report [6] emphasizes the issues of key exchange in symmetric key cryptography and the public key infrastructure (*PKI*) in asymmetric key cryptography. Hence, key management issue must be considered in the design of the protocol.
- The protocol should enable consumers to have the security control over his/her home, *i.e.*, control over all the home appliances with the smart meter.
- The protocol must support secure communication over the network with strong encryption. Moreover, the identity of each device should be protected over the network to maintain identity anonymity and untraceability.
- The protocol must be able to defeat various well-known security attacks, such as *MITM* attacks, redirection attacks, impersonation attacks, replay attacks, and flood-based *DoS* attacks.

1.1 Research Problem

There are several challenges with the current authentication protocols in terms of efficiency, overhead, cost, delay, privacy, and etc. Also, many vulnerabilities do exist in the available authentication schemes of various communication protocols for the *SG*, such as weak encryption and message digest in the *OSGP* protocol, security issues in the *DNP3* protocol (even in version 5), etc. There is not yet an integrated distributed authentication protocol that provides mutual authentications between the home environment (*HA*, *SM*, *HAN-GW*), energy provider, gateways (*BAN-GW*, *NAN-GW*), and *AMI* network (*SM*, aggregator/collector, *CC*). An integrated protocol can provide a common platform for authenticating various devices while efficiently maximizing the utilization of shared resources with low overhead in the *SG* system. Also, the privacy protection in the *SG* system is an important requirement, so the protocol must not reveal the confidential and private information related to any entity involved in the authentication process. Therefore, an end user (consumer) should have a control over his/her own home environment, such as *HA*, since data generated and being sent belong to a particular user. Furthermore, the protocol must be fast and efficient, and should be able to defeat known security attacks.

1.2 Our Contribution

In this paper, we design an integrated distributed protocol for the *SG* network, which meets all aforementioned challenges discussed in the previous subsection. Note that the proposed protocol may not be suitable for some parts of the *SG* system with very low communication latency requirements, such as for the generic object oriented substation event (*GOOSE*) and sampled measured values (*SMV*) layer-2 messages within the substation. Here, messages are not encrypted due to the transmission requirements within 4 milliseconds (*ms*). In such scenarios, a virtual *LAN* (*VLAN*) with layer-2 capabilities can be used with signed authenticated values [16], or a simple lightweight protocol can be designed for the authentication with integrity, but without encryption. Our new *SG* authentication protocol has following features:

- Provides mutual authentications between the *EP* and the *SM*, between the *SM*/*HAN-GW* and the *BAN-GW*/*NAN-GW*, between the *SM* and the *HA*, and between the *NAN-GW* and the *CC*.
- Provides a secure solution for the consumers to choose or change the energy provider of their own choice. The protocol provides more satisfaction to the consumer as he/she will have the control over its *HA* (secured with a password shared between the *SM* and all the *HA* and only he/she can change it).
- Defeats various security attacks: defeats the flood-based *DoS* attacks targeting transmitted messages between the *SM*/*HAN-GW* and the *BAN-GW*/*NAN-GW*; protects the *SM* and the *EP* from redirection attacks as Zip codes are verified at both ends; preserves the privacy of each message as it is encrypted before being transmitted over the network; provides resistances against *ID* thefts, *MITM* attacks, replay attacks, brute-force attacks, repudiation attacks, and impersonation attacks.
- Lightweight in terms of communication and computation overheads. The execution time of 3.96 s in Java can be considered fast, as it is for all the involved entities in the *SG* network and is within the requirements (few minutes) set by the standards [6].
- Uses the cloud-based trusted authorities (*TA*) for key management, which does not have the key exchange or *PKI* issues. Instead, the *TA* generates partial public and private keys, and the legitimate device generates its actual public and private keys.

1.3 Organization of the Paper

The rest of the paper is organized as follows: Section 2 discusses the related work and Section 3 presents our *SG* system model. A new authentication protocol is proposed in Section 4. Security and performance analysis is presented in Section 5, including a formal proof of the protocol. Section 6 presents the conclusion of this work.

Table 1 summarizes various symbols and abbreviations used in the paper along with their descriptions and sizes. Note that the sizes of public and private keys may depend upon the algorithm used in asymmetric encryption.

TABLE 1: Symbols And Abbreviations

Symbol	Description	Size (bits)
$H_1()/H_2()$	Hash functions used in ciphering	—
$H_3()$	Hash function for *SK* key generation	—
$H_{3_{change}}()$	Hash function for changing the password	—
$h()$	Hash function for computing e	—
ID	Identity of the entity	128
e	Hash value	128
MAC	Message authentication code	64
PUK	Public key	160
PRK	Private key	160
SK	Shared secret key	256
T	Timestamp	64
K	Random number	128
Zip	Postal code	128
S	Signature	128
pwd	Password shared between *SM* and *HA*s	128
Z	Sum of products of *K* and *ID*	128
P	Sum of products of *PRK* and *ID*	128
R	Sum of products of *S* and *ID*	128

2 Related Work

We first discuss the standardized protocols in SG along with their limitations. Then, we will present existing authentication protocols that provide authentications between various entities with lower overhead, and then those that provide protection against various security attacks and preserves the privacy in the *SG*.

There are some standardized protocols available in the literature for the *SG*, which support authentication process, such as open smart grid protocol (*OSGP*) for the smart meters, distributed network protocol (*DNP3*) between the control center and the substations, device language message specification/companion specification for energy metering (*DLMS*/*COSEM*) for the *AMI* network, and *OpenADR* for the demand response program. In addition, some standardized authentication protocols also support authentication, such as remote authentication dial-in user service (*RADIUS*) and *Diameter* protocols for the *2G*, *3G*, and *4G* cellular networks [9].

The *OSGP* protocol, which was developed by energy service network association (*ESNA*) and is a standard of the European telecommunications standards institute (*ETSI*), was deployed for providing the authentication and confidential security to the *SG* applications. This protocol is expected to provide reliable and efficient delivery of command and control information between the smart meters, direct load control modules, gateways, and other *SG* devices. However, recently, researchers from Germany easily recovered private encryption keys of the smart meters in a system following *OSGP* without a significant computational effort [10]. Also, a number of attacks has been performed over the *OSGP* protocol [11], including one with just 13 queries to a homegrown message authentication code (*OMA* digest) oracle, and by which the protocol further failed to deliver authenticity guarantee and confidentiality (due to using a non-standard composition of *RC4* as weak encryption algorithm) [10]. Similar security issues were found in the *DNP3* protocol, which does not provide authentication, message integrity, and confidentiality. In 2012, a new version of the *DNP3* protocol, named *DNP3* secure authentication version 5, was announced, which provides methods to remotely change user update keys using either symmetric or asymmetric cryptography [12]. However, it considers only spoofing, modification, and replay attacks over the network, and does not provide confidentiality of the message. Also version 5 of the protocol is not backwards compatible with previous versions, which may add a heavy protocol replacement cost.

Furthermore, the authentication supports provided by *DLMS*/*COSEM*, *OpenADR*, *RADIUS*, and *Diameter* are not sufficient and some of these are also very costly [9]. The *DLMS* (application layer communication protocol) and *COSEM* (data model), together provide an interface model for metering applications belonging to *IEC 62056* standards, such as electricity [13]. Basically, there are three authentication options in *DLMS*/*COSEM*, *i.e.*, no security, low level security authentication where server identifies client by a password, and high level security authentication

(mutual identification) with exchange challenges. However, the *DLMS/COSEM*'s security services are restricted to use symmetric key encryption. In practice, smart meters need asymmetric key to be used in secure socket layer/transport layer security (*TLS/SSL*), but *DLMS/COSEM* does not support *TLS/SSL*. In demand response, *OpenADR*, which is a standard development effort, supports authentication based on public key cryptography with exchange of certificates [14]. This standard maintains a hierarchy of certified authorities and requires a *PKI* to use three-tier *PKI* technology, which ultimately results in high development cost.

RADIUS is commonly used protocol to provide remote user authentication and accounting in cellular networks, and *WLAN* interworking and *Wi-Fi* offload situations [15]. It provides centralized services using a central database. However, the *SG* requires decentralized solutions, as a single-point-of-failure can massively affect the centralized system. *RADIUS* implementation supports peer authentication between communication endpoints using a pre-shared key. Hence, it brings some key management issues and is not suitable for large systems, such as the *SG*. Furthermore, *RADIUS* has poor scalability and uses the user datagram protocol (*UDP*), which does not provide reliable data transfer. Therefore, it is not suitable for the *SG* where the availability of information is extremely important. On the other hand, *Diameter* protocol is an authentication, authorization, and accounting protocol used in networking, which supports transmission control protocol (*TCP*) instead of *UDP*. However, its supported capabilities are sometimes more expansive. Furthermore, *RADIUS* and *Diameter* protocols do not directly protect against *DoS* attacks carried out by flooding the target equipment with bogus traffic.

For providing low overhead, a lightweight authentication scheme based on the Diffie-Hellman key exchange protocol and a hash-based message authentication code (*HMAC*) was proposed in [1]. However, it provides mutual authentication only between the *HAN-GW* and the *BAN-GW*. Sule *et al.* [17] made a change in [1] by using a *MAC* between the *AMI* devices and the controller nodes instead of *HMAC*. Although this scheme reduces the verification time, it also reduces the protocol security provided by the function. As in [1], the scheme only involves the *HAN-GW* and the *BAN-GW* communication. Further, an authentication scheme using a batch signatures verification was proposed in [18]. However, the scheme does not focus on authentication among *SM*, *HAN*, and *HA*, rather authenticating data aggregation. A key agreement protocol for *SG* is proposed in [19], which reduces the number of hash functions used and the delay caused by the security process. Recently, an identity-based scheme is proposed to provide authentication between the *SM* and the *AS*, and reduce the total number of exchanged packets, but increases the computation overhead [20].

Many researchers have proposed solutions in order to resist against different attacks in the *SG* system, such as replay, *MITM*, impersonation, and *DoS*. However, in the absence of authentication, an attacker can easily tamper the message and/or can send a fake message. In this direction, a mutual authentication scheme between the *SM* and the data concentration unit (*DCU*) was proposed to prevent impersonation, and *MITM* attacks [21]. However, this scheme neither discusses the generated overhead nor provides authentication in a home environment. Recently, an authentication scheme using a Merkle hash tree technique was proposed to prevent replay, injection and message modification attacks [22]. However, communication only between the *HAN* and the *NAN* is considered. A Diffie-Hellman-based secure aggregation scheme for collecting data was presented in [23], which generates lower computation and communication overhead, but it does not consider *SM*'s authentication. Metke *et al.* [24] stated that a strong authentication technique is required for all users and devices within the *SG* network. It is expected that in the near future, due to the increase in the number of devices, the current protocols may not be scalable.

In addition, the privacy of the customers in terms of power usage, billing, and other information must be preserved during the authentication. In this direction, an identity-based authentication protocol is proposed to provide source authentication, data integrity, non-repudiation services, and privacy preservation in *AMI* [25]. However, it does not consider overhead and efficiency. Yan *et al.* [26] proposed an integrated authentication and confidentiality (*IAC*) protocol that provides a mutual authentication between the *SM* and the *AMI* network, and enables data privacy, integrity, and confidentiality. However, it generates a huge overhead as it performs several encryption/decryption operations. Further, it does not consider *EP* and *HA* entities in their authentication system.

In summary, several standard, lightweight, and privacy-preserved protocols have been proposed by researchers. However, the existing standard protocols do not provide sufficient security and privacy preservation to the *SG* system. Also, many existing protocols (including privacy-preserved) are inefficient and generate huge overheads. Furthermore, the existing lightweight and privacy-preserved protocols are with limited capability of authenticating only few entities (mostly two devices) in the smart grid. In other words, these protocols do not enable authentications among all the entities with optimized resource utilization. Moreover, embedding security to the existing protocols generate huge overheads and requires integration to authenticate all the entities of the *SG* network, which results in inefficient and costly solution. Therefore, there is a need of an integrated lightweight authentication protocol that provides mutual authentications from end-to-end and protects the *SG* system from known attacks and keeping the privacy preserved. We tackle this problem in the paper.

3 System Model

Currently, in the *SG* system, security operations are done independently by each center. However, due to the limited processing capability, they do not support online analysis and generate high maintenance cost [27]. Further, the *SG* requires a powerful platform with effective integration and ubiquitous seamless access to collect and analyze huge data collected from a variety of sources such as *AMI*, wide area measurement system (*WAMS*), and *HA*. Recent studies [28], [29], [30], [31] show that cloud computing is very much compatible with the *SG* system because of its several advantages, including energy efficiency, flexibility, scalability,

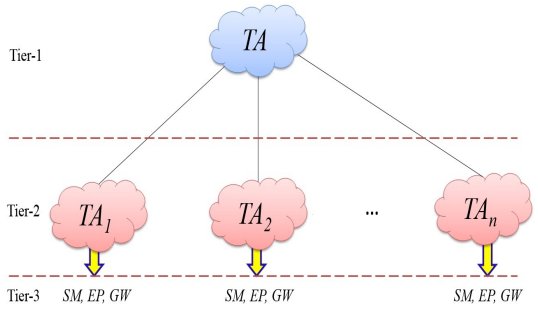


Fig. 2: Hierarchy of trusted authorities.

agility, and cost effectiveness. Various researchers have proposed their solutions by integrating cloud computing in the *SG* system. Baek *et al.* [29] designed a big data information management framework, called Smart-Frame, based on a cloud computing model. Also, Jiang *et al.* [32] proposed a scheme for searchable encryption on the cloud database in the *SG*, and Bitzer *et al.* [33] presented the feasibility study of monitoring renewable energy in the *SG* based on a cloud computing framework retaining *SG* security. Developing a secure cloud network is not our goal in this paper. However, we consider that our scheme uses secure cloud servers as discussed in [29], [32], and [33]. We employ the cloud computing into our *SG* system, particularly [29], which builds a hierarchical structure of cloud computing centers. Employing cloud computing in smart grids, not only addresses the issue of large information management, but also provides a high energy and cost saving platform. A roadmap in [34] presents a realistic example of deploying cloud computing centers in the *SG* system.

We propose to have a cyber-security layer on the top of communication protocols layer that takes care the security issues existing in the communication between any two entities over the network. Our *SG* system is divided into several regions/areas, each of which is managed by either a public or private, but secure cloud computing center [29]. As shown in Figure 2, we consider three different tiers in our *SG* system as follows:

- Tier-1: Central cloud computing center
- Tier-2: Distributed cloud computing centers
- Tier-3: *SM*s, *GW*s, and *EP*s

As shown in Fig. 2, there are n distributed cloud computing centers, also called trusted authorities (*TA*s). Each *TA* manages a region that includes various *SM*s, *GW*s, and *EP*s. The tier-1 *TA* provides inter-*TA* communication among different entities within the system, while the tier-2 *TA*s are responsible for managing the public key repository, and generating partial public and private keys of devices at their ends. The main purpose of enabling cloud environment in our *SG* system is to provide an easy and fast access to the public key repository and to efficiently generate public and private key pairs. In addition, the *SG* requires a powerful computing platform to handle a large scale data analysis and to support complex real-time application services. In each *TA*, various cloud computing services can be deployed, such as infrastructure-as-a-service (*IaaS*) for *SG* information collection, processing, and storage, platform-as-a-service (*PaaS*) for developing and integrating cloud computing specific security-based applications for the *SG* environment, and software-as-a-service (*SaaS*) for specific services such as optimization of energy usage.

4 Proposed Authentication Protocol

This section proposes an authentication protocol for the *SG* system. We first present an overview of our protocol, then present mutual authentication approaches between different *SG* entities. The authentications between *EP-SM*, *SM-GW*, and *SM-HA* are based on asymmetric key cryptography, asymmetric key cryptography in batch, and symmetric key cryptography, respectively.

4.1 Overview

Recently, identity-based cryptography (*IBC*) is considered to be a good platform for securing grid and cloud computing environments [35], [36]. However, *IBC* suffers from the key escrow problem [37]. Therefore, our protocol is based on a certificateless cryptosystem, which is a combination of identity-based cryptography and traditional public key cryptography [38]. Our approach not only overcomes the key escrow problem in *IBC*, but also does not require traditional *PKI* that is costly due to the private key generation (*PKG*) in *IBC*. We instead use a key generation center (*KGC*). The security of our scheme is based on the elliptic curve discrete logarithm problem (*ECDLP*) for the group of points on an elliptic curve over a finite field under elliptic curve cryptography (*ECC*). Here, we let E be an elliptic curve defined over a finite field F_p as $E : y^2 = x^3 + Ax + B; A, B \in F_p$. Let E_1 and E_2 be points in $E(F_p)$ and integer x is found such that $E_1 = xE_2$. The best known algorithm to solve the *ECDLP* is exponential, which is not enough to break its security. We do not design a pairing based scheme under *ECC*, but design a certificateless-based asymmetric encryption scheme. This is because the multiplication of points under *ECC* is more efficient than the pairing operation. For instance, it takes 0.6 *ms* for point multiplication and 4.5 *ms* for a pairing operation under a same setting [39]. The identity (*ID*) of each device (*EP*, *SM*, *GW*, *HA*) in the *SG* network is taken from a random point on the elliptic curve over $E(F_q)$.

Each *TA* generates its private and public key pair, known as a master private key and a master public key, and makes the public key available to its users. Our approach is simpler than the Diffie-Hellman protocol, as it uses one-way hash functions instead of exponential functions. The *KGC* (at each *TA*) supplies an entity with a partial private key and a partial public key. We assume that the *KGC* securely delivers the partial keys to the intended entities. The entity then combines its partial public and private keys with secret information to generate its actual private and public keys. In this way, the entity's private key is not known to the *KGC* and the anonymity of the user's public key is also achieved. This anonymity is useful, when we consider that in order to receive the public key of a device, the requested device must be verified authentic to the *TA* using its partial key credentials.

First, we present generic definitions of various algorithms used in our scheme, and then explain each of these

algorithms in detail.

Definition 1. A generic certificateless public key encryption scheme consists of the following algorithms:

- *Setup:* The *KGC* generates a common public parameter (*param*) and a master secret key (*masterKey*), and uses these keys to generate different keys.
- *PartialKeyGeneration: TA* uses *param*, *masterKey* and an identity *ID* (a point of elliptic curve group) received from a user to generate a partial private key (*PPR*) and a partial public key (*PPU*) as (*PPU*, *PPR*) = PartialKeyExtract (*param*, *masterKey*, *ID*).
- *SecretValue:* User/device uses *param* and *ID* to generate a secret value *sID* = SecretValue(*param*, *ID*).
- *GenPrivateKey:* User/device uses *param*, *PPR*, and *sID* to generate private key *PRK* as *PRK* = GenPrivateKey(*param*, *PPR*, *sID*).
- *GenPublicKey:* User/device uses *param*, *PPU*, *sID* and *ID* to generate public key *PUK* as *PUK* = GenPublicKey(*param*, *PPU*, *sID*, *ID*).
- *Encrypt:* The plaintext M is encrypted using *param* and *PUK* to generate a ciphertext C as C = Encrypt(*param*, *PUK*, M).
- *Decrypt:* The ciphertext C is decrypted using *param* and *PRK* to retrieve the plaintext M as M = Decrypt(*param*, *PRK*, C).

The public key of each entity is available in a public repository of the corresponding tier-2 cloud computing center (*TA*). The private keys are kept secret and stored on the *SM*s, the *GW*s, and the *EP*s. Since each entity is registered to a specific *TA*, it knows the identity and the public key of the *TA*. The details of generating different keys are as follows:

- *Setup:* $t \leftarrow \mathbb{Z}_q^*$ is a random integer with large prime q, and P is a generator of a large cyclic group G over $E(F_q)$. Each *TA* generates its private and public key pair as ($PRK_{TA} = t$, $PUK_{TA} = tP$) . Let us define the hash functions used in this protocol as $H_1 : \mathbb{Z}_q^* \rightarrow \{0,1\}^*$, $H_2 : \{0,1\}^* \rightarrow \{0,1\}^*$ and $H_3 : \mathbb{Z}_q^* \times \mathbb{Z}_q^* \rightarrow \{0,1\}^*$. Returns *param* = ($q$, P, PUK_{TA}, H_1, H_2, H_3) and *masterKey* = (q, P, PRK_{TA}, H_1, H_2, H_3).
- *PartialKeyGeneration: TA* chooses a random $s \in \mathbb{Z}_q^*$ and computes $w = sP$ and $x = s + PRK_{TA}H_1(ID)$. Note that the *ID* is first converted from a elliptic curve point to a bit string [40] in $H_1()$ and then hashed. Returns (*PPU*, *PPR*) = (w, x).
- *SecretValue:* Each device chooses a random $z \in \mathbb{Z}_q^*$. Returns *sID* = z.
- *GenPrivateKey:* Each device computes its private key *PRK* = (*sID*, *PPR*) = (z, x). Returns *PRK*.
- *GenPublicKey:* Each device computes its public key *PUK* = (*PPU*, *sID*, *ID*) = (w, v), where $v = zP$. Returns *PUK*.
- *Encrypt:* Sender device computes $r = H_2(M||\gamma)$, where $M \in \{0,1\}^*$ is a plaintext and $\gamma \in \{0,1\}^*$. Furthermore, it computes ciphertext $C = (c_1, c_2, c_3)$ such that $c_1 = rP$; $c_2 = rv + M||\gamma$; $c_3 = w + u$; where $u = PUK_{TA}H_1(ID)$. Returns C.
- *Decrypt:* Receiver device first applies partial private key by computing $Ver_1 = c_3 - xP$. If $Ver_1 = 0$, it proceeds further, otherwise terminates the connection. Thereafter, it retrieves the message $M||\gamma$ as $c_2 - zc_1$ and verifies Ver_2 as $H_2(M||\gamma)P \stackrel{?}{=} c_1$. Returns M.

4.2 Authentication between the EP and the SM

We assume that *EP* knows the identity of each *SM* that it supplies the electricity to. Similarly, each *SM* also knows the identity of its *EP*, as it has a contract with the *EP*. As shown in Figure 3, the authentication between the *EP* and the *SM/HAN-GW* is carried out as follows:

Step-1 ***EP*→*SM*:**$[E_{PUK_{SM}}\{ID_{EP}, K_1, Zip_{EP}\}, T_1, MAC_1]$**:** First, the *EP* retrieves the public key of the *SM* from the repository stored at its tier-2, *i.e.*, PUK_{SM}. Then, it encrypts its identity ID_{EP}, a nonce K_1, and the location (Zip code) Zip_{EP} with the public key of the *SM* and sends it to the *SM* along with a current timestamp T_1 and a MAC_1 (message-1), where $MAC_1 = [E_{PUK_{SM}}\{ID_{EP}, K_1, Zip_{EP}\}, T_1]$. We consider each *MAC* as a *HMAC* function, *i.e.*, *HMACSHA256*, and a pre-assigned key, say K, is used in *MAC*.

Step-2 ***SM*→*EP*:**$[E_{PUK_{EP}}\{ID_{SM}, K_2, Zip_{SM}\}, T_2, MAC_2]$**:** On receiving message-1, the *SM* computes $MAC_1^{'}$ and checks if $MAC_1 \stackrel{?}{=} MAC_1^{'}$. If it is true, the *SM* decrypts the message using its private key. Then, the *SM* retrieves the public key of the *EP* (PUK_{EP}) and verifies the identity and the location of the *EP*. If it is true, then the *SM* sends ID_{SM}, K_2, Zip_{SM} encrypted with PUK_{EP} to the *EP* along with T_2 and MAC_2 (message-2), where $MAC_2 = [E_{PUK_{EP}}\{ID_{SM}, K_2, Zip_{SM}\}, T_2]$.

Step-3: On receiving message-2, the *EP* computes $MAC_2^{'}$ and checks if $MAC_2 \stackrel{?}{=} MAC_2^{'}$. If it is true, the *EP* decrypts the received message using its private key, and verifies the identity and the location of the *SM*. If both are correct, the *EP* computes a secret shared key as $SK_1 = H_3(Zip_{EP} \oplus K_2, Zip_{SM} \oplus K_1)$ and sends message to the *SM* encrypted with this shared key. Here, $H_3()$ is a one-way hash function. Similarly, the *SM* also computes the same secret SK_1 key.

4.3 Authentication between the SM and the GW

We assume that a group of *SM* sends its metering data to a specific *GW*. The *GW* keeps a record of the identity of each *SM* associated with it. A number of *SM* communicates with a *GW* simultaneously, so the mutual authentication is executed in a batch. The authentication process and

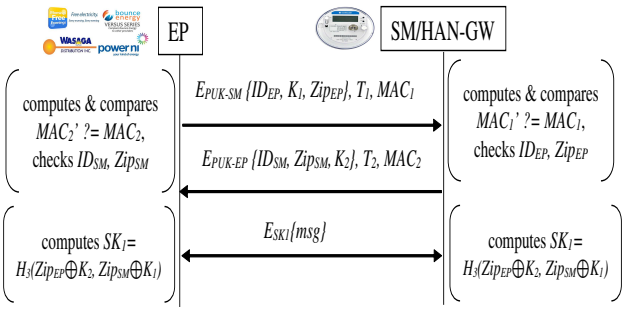


Fig. 3: Authentication between the EP and the SM.

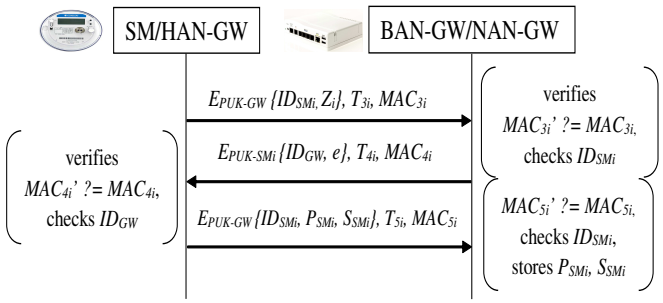


Fig. 4: Authentication between the SM and the BAN/NAN-GW.

the communication scenario of the proposed authentication scheme between a group of *SM*s and the *GW* are shown in Figure 4 and Figure 5, respectively. As shown in Figure 4, the authentication process is carried out as follows:

Step-1 $SM_i \rightarrow GW$:$[E_{PUK_{GW}}\{ID_{SM_i}, Z_i\}, T_{3_i}, MAC_{3_i}]$: First, each SM_i retrieves the identity and the public key of the *GW*. Then, each SM_i sends its identity and Z_i encrypted with PUK_{GW} along with its current timestamp T_{3_i} and MAC_{3_i} to the *GW* (message-1), where $MAC_{3_i} = [E_{PUK_{GW}}\{ID_{SM_i}, Z_i\}, T_{3_i}]$ and $Z_i = ID_{SM_i}K_i$. The $K_i \in [1, q\text{-}1]$ are the random secret values selected by each SM_i.

Step-2 $GW \rightarrow SM_i$:$[E_{PUK_{SM_i}}\{ID_{GW}, e\}, T_{4_i}, MAC_{4_i}]$: On receiving message-1, the *GW* computes MAC'_{3_i} and checks message integrity. If it is true, the *GW* compares its current timestamp t_m with $T_{threshold} = T_{3_i} + \epsilon$, where ϵ is the maximum allowed delay to transmit the message to the *GW*. If $t_m > T_{threshold}$, the request is discarded and the connection is terminated. Otherwise, the *GW* decrypts the message using its private key, and verifies the identity of the SM_i. If it is true, the *GW* checks the number of attempts by the SM_i within a specified interval. If it is more than the assigned limit, the connection is terminated. Otherwise, the *GW* sends its identity and a value e encrypted using the public key of the corresponding SM_i along with T_{4_i} and MAC_{4_i} (message-2) to the SM_i. Here, $e = h(Z)$, h is a one-way hash function, $Z = \sum_{i=1}^{n} Z_i$, and n is the number of SM_i communicating with the *GW*.

Step-3 $SM_i \rightarrow GW$:$[E_{PUK_{GW}}\{ID_{SM_i}, P_{SM_i}, S_{SM_i}\}, T_{5_i}, MAC_{5_i}]$: On receiving message-2, each SM_i computes MAC'_{4_i} and verifies the integrity of each message. If it is true, SM_i decrypts the messages using private keys PRK_{SM_i}, and verifies the received identity of the *GW*. If it is true, each SM_i stores e, and generates a variable $P_{SM_i} = PRK_{SM_i}ID_{SM_i}$ and a signature $S_{SM_i} = (K_i + ePRK_{SM_i}) \bmod n$. Note that the first 128 *bits* of $P_{PRK_{SM_i}}$ are used in P_{SM_i} and S_{SM_i} for operations' compatibility. Then, each SM_i sends ID_{SM_i}, P_{SM_i}, and S_{SM_i} encrypted using public key of the *GW* along with T_{5_i} and $MAC_{5_i} = [E_{PUK_{GW}}\{ID_{SM_i}, P_{SM_i}, S_{SM_i}\}, T_{5_i}]$ (message-3) to the *GW*. On receiving message-3, the *GW* computes MAC'_{5_i} and checks message integrity. If it is true, the *GW* decrypts the messages, and verifies the identity of each SM_i. In a scenario where a group of SM_i communicates with a *GW*, it is possible that some of them perform flood-based *DoS* attacks instead of sending message-3. In order to prevent these attacks, the identity of each SM_i is verified. For each unresponsive SM_i, the *GW* removes the corresponding Z_i and re-computes Z. Then, the *GW* computes $P = \sum_{i=1}^{n} P_{SM_i}$ and $R = \sum_{i=1}^{n} S_{SM_i}ID_{SM_i}$, and verifies ($R$ - $eP \stackrel{?}{=} Z$).

Therefore, our scheme is efficient even with the presence of invalid requests in a batch since the *GW* only needs to re-compute Z, which is simply a summation of all Z_i.

4.4 Authentication between the HA and the SM

Since data generated and sent by all the *HA* belong to a particular user, we involve the end user (owner) for authenticating the home appliances (at the initial setup) [41]. The energy consumption information can reveal personal details of the consumers, such as their daily routines (including times when they are at home or asleep), what electronic equipment they own and are being used, etc. Consumers expect that the privacy of this information is maintained. We assume that the *SM* and all the *HA* share a password selected by the user. A secret key $SK_2 = H_3(pwd, T)$ is generated every time a *HA* and the *SM* communicates, where pwd is the shared password, T is a timestamp, and H_3 is a one-way hash function. As shown in Figure 6, the authentication process between the *HA* and the *SM* is carried out as follows:

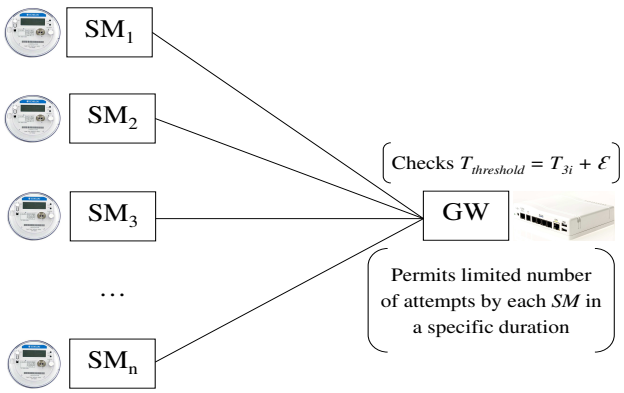


Fig. 5: Communication scenario between a group of SMs and the GW.

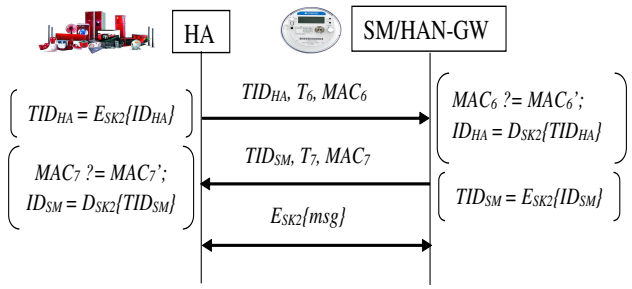


Fig. 6: Authentication between the HA and the SM.

Step-1 ***HA*→*SM*:**$[TID_{HA}, T_6, MAC_6]$**:** First, each *HA* generates SK_2 from a shared password and uses it to encrypt the original identity of the *HA*. Then, it sends a temporary identity TID_{HA}, a timestamp T_6, and MAC_6 to the *SM* (message-1), where $MAC_6 = [TID_{HA}, T_6]$ and $SK_2 = H_3(pwd, T_6)$. The encryption can be performed by any standard symmetric key algorithm such as *AES-CTR* or *MAES-CTR* [42].

Step-2 ***SM*→*HA*:**$[TID_{SM}, T_7, MAC_7]$**:** On receiving message-1, the *SM* verifies $MAC_6^{'}$ with the received MAC_6. If it is true, the *SM* decrypts and recovers the actual identity of the *HA*. If the identity belongs to one of its *HA*, it generates a temporary identity TID_{SM} and sends its identity to the *HA* along with T_7 and MAC_7 (message-2), where $MAC_7 = [TID_{SM}, T_7]$.

On receiving message-2, the *HA* computes $MAC_7^{'}$ and compares it with MAC_7, and further decrypts and recovers the actual identity of the *SM*. If it is correct, the *HA* and the *SM* can start communicating using messages encrypted by SK_2. Moreover, the password can be automatically changed at a regular interval by calculating $pwd_{i+1} = N \times H_{3_{change}}(d \times pwd_i)$, where N is the number of days, d is a random secret, and $H_{3_{change}}$ is a hash function. For the password change, the user needs to provide N to the *SM*. When, a new password is generated at *SM*, the *SM* encrypts it using last session key and sends it to all the *HA* before discarding the previous key.

4.5 Authentication between the NAN-GW and the CC

We assume that the *NAN-GW* aggregates the received data from various *SM*. The *CC* is assumed to be connected to the *NAN-GW* using wired network and is authenticated. In case, if it is wireless connected, the scenario similar to *EP-SM* provides mutual authentication.

5 Security and Performance Analysis

This section presents the verification proofs, defenses against various security attacks, and security and performance analysis of our protocol in comparison with the existing lightweight protocols.

5.1 Verification Proof

We present the verification proofs for the public decryption of our public encryption scheme, and the correctness of the protocol between *SM*s and their corresponding *GW*.

1. Verification of Decryption in Our Encryption Scheme:

$$\begin{aligned} Ver_1 &= c_3 - xP \\ &= w + u - xP \\ &= sP + PUK_{TA}H_1(ID) - [s + PRK_{TA}H_1(ID)]P \\ &= sP + PRK_{TA}H_1(ID)P - sP - PRK_{TA}H_1(ID)P \\ &= 0. \end{aligned}$$

$$\begin{aligned} Ver_2 &= H_2(M||\gamma)P \stackrel{?}{=} c_1 \\ &= H_2(M||\gamma)P \stackrel{?}{=} rP \\ &= H_2(M||\gamma)P \stackrel{?}{=} H_2(M||\gamma)P. \end{aligned}$$

2. Correctness of the Protocol between *SM*$_i$*-GW*:

$$\begin{aligned} L.H.S. &= Z \\ &= \sum_{i=1}^{n} Z_i \\ &= ID_{SM_1}K_1 + ID_{SM_2}K_2 + ... + ID_{SM_n}K_n. \\ R.H.S. &= R - eP \\ &= (S_{SM_1}ID_{SM_1} + S_{SM_2}ID_{SM_2} + ... + S_{SM_n} \\ &\quad ID_{SM_n}) - e(P_{SM_1} + P_{SM_2} + ... + P_{SM_n}) \\ &= ((K_1 + e(PRK_{SM_1}))ID_{SM_1} + (K_2 + e \\ &\quad (PRK_{SM_2}))ID_{SM_2} + ... + (K_n + e \\ &\quad (PRK_{SM_n}))ID_{SM_n}) - e(P_{SM_1} + P_{SM_2} + \\ &\quad ... + P_{SM_n}) \\ &= (ID_{SM_1}K_1 + ID_{SM_2}K_2 + ... + ID_{SM_n}K_n) + e \\ &\quad ((PRK_{SM_1})ID_{SM_1} + (PRK_{SM_2})ID_{SM_2} \\ &\quad + ... + (PRK_{SM_n})ID_{SM_n}) - e(P_{SM_1} + \\ &\quad P_{SM_2} + ... + P_{SM_n}) \\ &= (ID_{SM_1}K_1 + ID_{SM_2}K_2 + ... + ID_{SM_n}K_n) + e \\ &\quad (P_{SM_1} + P_{SM_2} + ... + P_{SM_n}) - e(P_{SM_1} + \\ &\quad P_{SM_2} + ... + P_{SM_n}) \\ &= ID_{SM_1}K_1 + ID_{SM_2}K_2 + ... + ID_{SM_n}K_n = Z. \end{aligned}$$

5.2 Defenses Against Various Security Attacks

We assume that an adversary $\mathcal{A}$ has a complete knowledge about the system topology, as well as the identities and public keys of the entities. $\mathcal{A}$ may be an internal entity or an external entity. $\mathcal{A}$ may attempt to launch *MITM* attacks on the active connections between any two entities of the *SG* network. Since all messages over the network are encrypted, inherently, the *MITM* attacks will not be successful. The replay attacks are also prevented as every message over the network contains a unique timestamp value. As discussed in Section 4.3, the proposed protocol also defeats the flood-based *DoS* attacks. In addition, the impersonation attacks can be prevented, since the fake request is discarded and the connection is terminated. $\mathcal{A}$ does not have the actual private key/shared secret key of the valid entity and cannot decrypt the transmitted message. The key size of each shared secret key and public key/private key is chosen to be longer than 128 *bits* to resist against brute-force attacks. Furthermore, the Zip codes, sent by the devices, are used to overcome the redirection attacks. Table 2 shows the comparison of the security capabilities of the proposed protocol with the existing protocols. Note that [19] and [20] partially protect *DoS* attacks by simply limiting the key agreement sessions.

TABLE 2: Behavior of Attacks in Various Authentication Protocols

Vulnerabilities	[19]	[1]	[20]	Proposed
MITM attacks	Yes	Yes	Yes	Yes
Replay attacks	Yes	Yes	Yes	Yes
Impersonation attacks	Yes	Yes	Yes	Yes
Brute-force attacks	Yes	Yes	Yes	Yes
Redirection attacks	No	No	No	Yes
Flood-based DoS attacks	Partial	No	Partial	Yes

5.3 Security Analysis

The proposed protocol provides mutual authentications between the *EP* and the *SM*, between the *SM* and the *GW*, and between the *SM* and the *HA*. Our protocol also provides a *perfect forward secrecy*, since the adversary can neither retrieve the actual key nor predict any of the future keys using any shared secret key. Furthermore, our protocol *preserves the privacy* of communicated entities over the network and *overcomes ID thefts*, since the transmitted messages are always encrypted. Table 3 shows the comparison of security requirements. Note that we have a system with $|K|=|C|=|P|$, each of 128 *bits* (with *AES-CTR*) or 256 *bits* (with *MAES-CTR*) for symmetric encryption and $|K|\geq|C|=|P|$ for asymmetric encryption. Therefore, our system has perfect secrecy as each key is used with equal probability $1/|K|$, and for every plaintext P and ciphertext C there is a unique key K such that $E_K(P) = C$. As well, our system with at least equal size spaces $|P|=|C|=|K|$ is *perfectly key ambiguous* as the keys are picked uniformly, and for all $x \in P$, $y \in C$, there is a unique key K such that $y = E_K(x)$.

Furthermore, $\mathcal{A}$ cannot retrieve the partial and actual private keys of any device. Even in other scenarios where $\mathcal{A}$ extracts any one of these parameters (i) partial private key, (ii) partial public key, and (iii) public key, or replaces the public key of the device, our public encryption scheme is able to defend such attacks, as $\mathcal{A}$ cannot retrieve the actual private key and cannot decrypt the message. Let us consider two scenarios, in which $\mathcal{A}$ tries to extract some information:

Scenario-1: $\mathcal{A}$ does not have access to the *masterKey*, but may replace public keys (*PUK*) of the devices with any value, and also requests the public key of victim device, extracts the partial private key, and makes decryption queries. Under this scenario, $\mathcal{A}$ has following restrictions:

- $\mathcal{A}$ cannot extract the partial private key (*PPR*) of the challenge device *ID* at any point, as the fake *ID* will be discarded by the *TA*.
- $\mathcal{A}$ cannot request the private key (*PRK*) of any identity, if the respective public key (*PUK*) has been replaced.
- $\mathcal{A}$ cannot make a decryption query on the challenge ciphertext C that was generated by a combination of (*ID*, *PUK*).

Scenario-2: $\mathcal{A}$ does have access to the *masterKey*, but may not replace public keys (*PUK*) of the devices. $\mathcal{A}$ can compute partial private key of any device, and also can request public key and make private key extraction and decryption queries. Under this scenario, $\mathcal{A}$ has following restrictions:

TABLE 3: Summary of Security Requirements Fulfilled by Various Protocols

Requirements	[19]	[1]	[20]	Proposed
Mutual authentication	Yes	Yes	Yes	Yes
Forward secrecy	Yes	Yes	Yes	Yes
Privacy preservation	No	Yes	No	Yes
Prevents ID thefts	No	Yes	No	Yes

- $\mathcal{A}$ cannot replace the public key (*PUK*) of any device at any time, as the identity and public key repositories are stored at various trusted authorities (*TA*).
- $\mathcal{A}$ cannot extract the private key (*PRK*) of the challenge device at any time, as it is randomly selected by each device.
- $\mathcal{A}$ cannot successfully decrypt the challenge ciphertext C on behalf of the victim device, as it may generate partial private key (*PPR*) of the device, but does not have the actual private key (*PRK*) of the device.

Definition 2. A protocol is secure against adaptive chosen plaintext attack (*IND-CPA* secure), if no polynomially bounded adversary has a non-negligible advantage. Therefore, our protocol is secure against *IND-CPA*.

Our system is secure in terms of indistinguishability as $\mathcal{A}$ cannot identify the message choice because of a unique combination of P and K for each transmitted message C. Here, *Indistinguishability under chosen plaintext attack* (*IND-CPA*) is equivalent to the property of *semantic security*. In our protocol, symmetric encryption is performed by *AES-CTR*, which is *IND-CPA* secure. Also, the asymmetric encryption, performed by our proposed scheme, is based on *ECC* and is indistinguishable under chosen ciphertext attack (*IND-CCA* secure), considering the hardness of the *ECDLP* [43].

5.4 Performance Analysis

A mutual authentication between the *HAN-GW* and the *BAN-GW* is proposed in [1], a mutual authentication between the *SM* and the *AS* of the *DCU-GW* is proposed in [20]. A number of authentication scenarios between *SM*, *HAN-GW*, *BAN-GW*, *NAN-GW*, and *HA* are presented in [19], whereas our protocol proposes mutual authentications between *EP*, *SM*, *HAN-GW*, *BAN-GW*, *NAN-GW*, and *HA*. This subsection computes and compares communication and computation overheads among these four protocols, and evaluates total execution time of our proposed protocol.

The total communication overhead (*CMO*) and the total computation overhead (*CPO*) of the protocol for a single authentication token are calculated, respectively, as $CMO_{total} = CMO_{EP-SM} + CMO_{SM-GW} + CMO_{SM-HA}$ and $CPO_{total} = CPO_{EP-SM} + CPO_{SM-GW} + CPO_{SM-HA} + CPO_{key-gen}$. Table 4 shows the comparison of the *CMO* and *CPO* of our protocol with the existing protocols [1], [19], [20]. Out

TABLE 4: Performance with Single Authentication Token

Performance Parameter	[19]	[1]	[20]	Proposed
Computation overhead	8*E*, 3*XOR*, 8*D*, 27*H*, 19*MUL*	3*E*, 3*D*, 2*H*, 2*HMAC*, 4*EXP*	13*H*, 3*MUL*, 2*XOR*, 1*ADD*, 1*SUB*, 4*EXP*	7*E*, 4*EMUL*, 7*D*, 1*ESUB*, 5*H*, 4*XOR*, 14*HMAC*, 1*MUL*, 1*ADD*
Communication overhead (bit)	3712	1152	1152	2752
Entities involved in authentication	*SM*, *HAN-GW*, *HA*, *BAN-GW*, *NAN-GW*	*HAN-GW*, *BAN-GW*	*SM*, *AS* of *DCU-GW*	*EP*, *SM*, *HA*, *HAN-GW*, *BAN-GW*, *NAN-GW*

of these three existing protocols, it is fair to compare our protocol with only the protocol in [19], as only this protocol includes most of the involved entities in the *SG*, while only two entities are involved in [1] and [20]. Although, the protocol in [19] and our protocol cover a similar range of entities, our protocol achieves much lower overhead. In detail, authentication scenario between the *EP-SM* generates *CMO* of 1024 *bits* and prevents *MITM*, replay, impersonation, and redirection attacks. The scenario between the *SM-GW* generates 1216 *bits* of *CMO* and prevents *MITM*, replay, impersonation, repudiation, and flood-based *DoS* attacks. In comparison with the protocol in [1], our protocol is also resistant against the flood-based *DoS* attacks while adding just 24 *bits* of *CMO*. Furthermore, in the authentication scenario between the *SM-HA*, our protocol prevents *MITM*, replay, impersonation, and brute-force attacks while generating 512 *bits* of *CMO*.

We also evaluate the performance of our protocol when there are multiple authentication tokens. We assume that there are m users executing the protocol simultaneously and each user has n home appliances. The *CMO* generated by the proposed protocol is calculated as $CMO(m,n) = CMO(EP{-}SM)m + CMO(SM{-}GW)m + CMO(SM{-}HA)n$ = 1024m+1216m+512n = 2240m+512n. The *CPO* generated by the proposed protocol is calculated as $CPO(m,n)$ = (5m+2n)*E*+(5m+2n)*D*+(3m+n+1)*H*+(10m+4n)*MAC*+1*ESUB*+4m*EMUL*+1*MUL*+m*ADD*+(2m-2)*EADD* +4m*XOR*. Here, *E* and *D* represent encryption and decryption, respectively; *XOR* is bit-wise exclusive-*OR*, *MUL* and *ADD* are scalar multiplication and addition over integers/binaries, respectively; *EMUL*, *EADD*, and *ESUB* are elliptic curve multiplication, addition and subtraction (all three are computed as additions), respectively; and *H* and *MAC* are hash and authentication code functions, respectively. Furthermore, we assume that there are r malicious users in a batch. The protocol first removes the invalid requests of the malicious users and then computes other parameters before further executing the protocol. In such case, the total recalculated *CPO* is as $CPO(m,n,r)$ = $CPO(m,n)$ - r*MUL* - 2r*EMUL* - r*ESUB* - 2(r-1)*EADD*. Since *XOR* operations are negligible in comparison with other operations, they are not included in the calculation of *CPO*.

Figure 7 and Figure 8, respectively, show the *CMO* and *CPO* generated by the proposed protocol for different number of users (m = 10, 50, 100) and the number of home appliances (n = 1, 5, 10, 20), considering unit value for each operation. In Figure 7, $CMO(10,1)$ = 2864 and $CPO(10,1)$ = 38.75 *bytes*, $CMO(100,20)$ = 29280, and $CPO(100,20)$ = 397.625 *bytes*. In Figure 8, $CPO(100,5,1)$ = 380 *bytes*, $CPO(100,10,50)$ = 348.875 *bytes*, and $CPO(100,20,99)$ = 323.375 *bytes* (worst case). Hence, even if there are some invalid requests r (Figure 8(b): 1, $m/2$, and m-1) in a batch, our protocol efficiently handles them.

5.5 Simulation Result

We simulated our protocol in Java environment with JDK1.7, 2GB RAM, and Windows7 OS. For a single authentication token, the scalar addition and multiplication operations over integer/binaries took 0.000933 milliseconds (*ms*) and 0.00918 *ms*, single addition and doubling over elliptic curve took 0.6031 *ms* and 0.6047 *ms*, hash function *SHA256* took 0.9 *ms*, *HMAC* function *HMACSHA256* took 271.60 *ms*, and [encryption, decryption] time of symmetric *MAES-CTR* mode with 256 *bits* key between *EP-SM* and *SM-HA* took (0.97, 0.78) *ms*. Moreover, the asymmetric encryption (i) using *RSA* with 2048 *bits* key and (ii) using certificateless public encryption scheme took (30, 16) *ms* and (12, 7.6) *ms*, respectively. The total computation time by our protocol using *RSA* and using proposed scheme is 4041.91 *ms* and 3962.71 *ms*, respectively. This computation time can be further reduced by using the fast multiplication, where a single addition and doubling takes approximately half of the ordinary *ECC* multiplication, *i.e.*, 0.303 *ms* [44]. The total messages (2752 *bits*) transmission time on *3G* and *4G* networks [45] by our protocol are 0.000451 and 0.000182 *ms*, respectively. Hence, the total execution time by our protocol (with certificateless cipher scheme) on *3G* and *4G* networks of approximately 3.96 *s* is quite reasonable, considering that it is the total time for completing authentications for all involved entities in the *SG* network. Here, we presented just one case for the overall protocol execution time. However, if we encrypt the message with *AES-CTR/MAES-CTR* for the asymmetric encryption, and the symmetric key is encrypted by an asymmetric algorithm, the overall time can be further reduced.

5.6 Formal Proof of the Properties of the Protocol

In order to justify our analysis, we use the *BAN-Logic* to provide a formal proof of our scheme. The notations used in BAN-Logic can be referred from [46].

1) Message Meaning Rule:

(1) $\frac{EP|\equiv(EP \overset{SK_1}{\leftrightarrow} SM), EP \triangleleft E\{ID_{MP}, K_1, Zip_{EP}\}_{PUK_{SM}}}{EP|\equiv SM|\sim E\{ID_{EP}, K_1, Zip_{EP}\}_{PUK_{SM}}}$

(2) $\frac{SM|\equiv(SM \overset{SK_1}{\leftrightarrow} EP), SM \triangleleft E\{ID_{SM}, K_2, Zip_{SM}\}_{PUK_{EP}}}{SM|\equiv EP|\sim E\{ID_{SM}, K_2, Zip_{SM}\}_{PUK_{EP}}}$

2) Timestamp Verification Rule:

(1) $\frac{SM_i|\equiv\#(T_i), SM_i|\equiv GW|\sim msg_1 \wedge msg_3}{SM_i|\equiv GW|\equiv msg_1 \wedge msg_3}$

(2) $\frac{GW|\equiv\#(T_j), GW|\equiv SM_i|\sim msg_2}{SM_i|\equiv GW|\equiv msg_2}$

3) Jurisdiction Rule:

(1) $\frac{HA|\equiv SM \Rightarrow TID_{HA}, HA \triangleleft HA|\sim TID_{HA}}{HA|\equiv SM}$

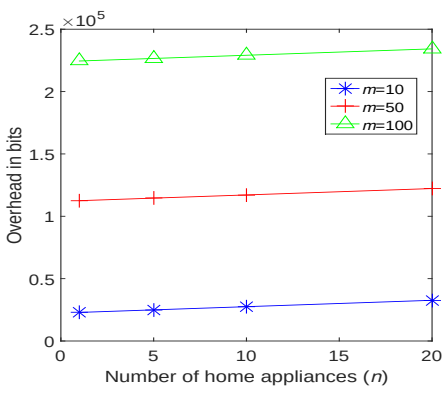
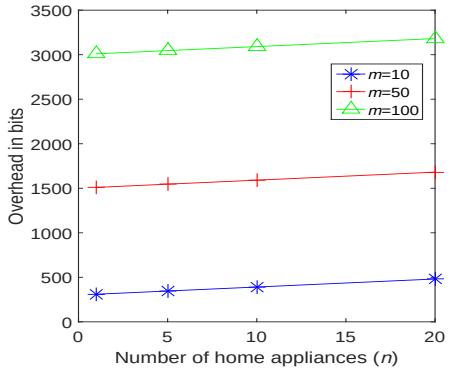
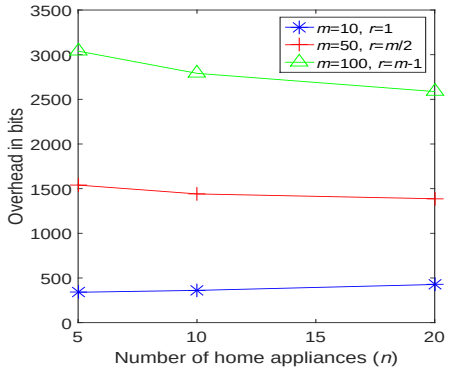


Fig. 7: Communication overhead.



(a) m = 10, 50, 100; n = 1, 5, 10, 20.



(b) m = 10, 50, 100; n = 5, 10, 20; r = 1, $m/2$, m-1.

Fig. 8: Computation overhead.

(2) $\frac{SM|\equiv HA \Rightarrow TID_{SM}, SM \triangleleft SM|\sim TID_{SM}}{SM|\equiv HA}$

4) Protocol Goals:

a. Mutual Authentication: $EP|\equiv SM \wedge EP \rightarrow SM|\equiv EP \wedge SM$. Thus, mutual authentication holds.

b. Session Key Agreement: Each key SK_1 between each *EP* and the *SM* provides session key agreement.

c. Freshness of messages: $SM|\equiv \#(T_j) \wedge EP|\equiv \#(T_i)$. Hence, freshness of messages between the *EP* and the *SM* holds.

d. Integrity and Privacy between the EP and the SM:

(1) $\frac{EP|\equiv (EP \overset{SK_1}{\leftrightarrow} SM), EP \triangleleft HMAC\{msg\}}{EP|\equiv SM|\sim msg}$

(2) $\frac{EP|\equiv (EP \overset{SK_1}{\leftrightarrow} SM), EP \triangleleft E\{ID\}_{SK_1}}{EP|\equiv SM|\sim ID}$

6 Conclusion

The proposed protocol, based on hierarchical cloud trusted authorities, provides mutual authentications between the *EP* and the *SM*, between the *SM*/*HAN-GW* and the *BAN-GW*/*NAN-GW*, between the *SM* and the *HA*, and between the *NAN-GW* and the *CC*. Particularly, the authentications between *EP-SM* and *GW-CC*, *SM-GW*, and *SM-HA* are, respectively, based on asymmetric key cryptography, asymmetric key cryptography in batch, and symmetric key cryptography. Processing requests in batch improves the efficiency of the system, as a large number of smart meters communicate with the gateway simultaneously for mutual authentications. The certificateless scheme in the proposed protocol maintains privacy preservation as the transmitted message is always encrypted over the network. Simulation results show that the authentication scenarios between the *EP-SM*, the *SM-GW*, and the *SM-HA* generate lower communication and computation overheads in comparison with existing protocols. Also the overheads generated by our protocol are manageable, even when invalid requests exist in a batch. Through security analysis, we show that our protocol is secure against various existing attacks, such as *MITM* attacks, replay attacks, impersonation attacks, redirection attacks, and flood-based *DoS* attacks. In sum, our protocol is lightweight with low execution time and efficiently provides a centralized integrated control in a decentralized environment. Furthermore, our protocol can be readily integrated with the cloud computing-based trusted entities to utilize powerful computing services of the cloud to efficiently manage the *SG* system.

Acknowledgments

This research was funded by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the "ICT Consilience Creative Program" (IITP-2015-R0346-15-1007) supervised by the IITP (Institute for Information & Communications Technology Promotion) and under the "Basic Science Research Program" (2013R1A1A1010489, 2015R1C1A1A01053788) through the NRF (National Research Foundation).

References

[1] M. Fouda, Z. Fadlullah, N. Kato, R. Lu, and X. Shen, "A lightweight message authentication scheme for smart grid communications," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 675-685, Dec. 2011.

[2] M. Balakrishnan, "Smart energy solutions for home area networks and grid-end applications," in *Proc. Smart Energy*, 2012, pp. 67-73.

[3] W. Wang and Z. Lu, "Cyber security in the smart grid: survey and challenges," *Computer Networks*, vol. 57, pp. 1344-1371, 2013.

[4] A. Bari, J. Jiang, W. Saad, and A. Jaekel, "Challenges in the smart grid applications: an overview," *International Journal of Distributed Sensor Networks*, vol. 2014, pp. 1-11, 2014.

[5] Smart Grid Cyber Security, Potential Threats, Vulnerabilities and Risks, California State Uni., May 2012. [Online]. www.energy.ca.gov/2012publications/CEC-500-2012-047/CEC-500-2012-047.pdf.

[6] Guidelines for Smart Grid Cyber Security, NISTIR7628, Aug. 2010. [Online]. csrc.nist.gov/publications/nistir/ir7628/nistir-7628_vol3.pdf.

[7] Report to NIST on the Smart Grid Interoperability Standards Roadmap, Electric Power Research Institute, Aug. 2009. [Online]. www.nist.gov/smartgrid/upload/Report_to_NIST_ August10_2.pdf.

[8] A. Al-Majali, A. Vishwanathan, and C. Neuman, "Analyzing resiliency of the smart grid communication architectures under cyber attack," in *Proc. CSET*, 2012, pp. 1-8.

[9] N. Saxena and B. J. Choi, "State of the art authentication, access control, and secure integration in smart grid," *Energies*, vol. 8, no. 10, pp. 11883-11915, Oct. 2015.

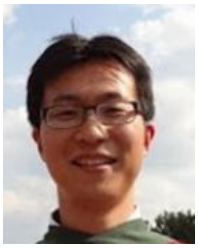
[10] P. Jovanovic and S. Neves, "Dumb crypto in smart grids: practical cryptanalysis of the open smart grid protocol," *IACR*, Apr. 2015. [Online]. https://eprint.iacr.org/2015/428.pdf.

[11] K. Kursawe and C. Peters, "Structural weaknesses in the open smart grid protocol," *IACR*, June 2015. [Online]. https://eprint.iacr.org/2015/088.pdf.

[12] DNP3 Secure Authentication Version 5, Apr. 2012. [Online]. https://www.dnp.org/Lists/Announcements/Attachments/7/ Secure Authentication v5 2011-11-08.pdf.

[13] IEC 62056-6-2:2013, Electricity Metering Data Exchange - The DLMS/COSEM Suite - Part 6-2: COSEM Interface Classes, 2006. [Online]. https://webstore.iec.ch/publication/6410.
[14] OpenADR and Cyber Security. [Online]. http://www.openadr.org/cyber-security.
[15] Remote Authentication Dial in User Service - RADIUS, Developing Solutions. [Online]. https://www.developingsolutions.com/products/radius.
[16] IEC TS 62351-6:2007, Power Systems Management and Associated Information Exchange - Data and Communications Security - Part 6: Security for IEC 61850. [Online]. https://webstore.iec.ch/publication/6909.
[17] R. Sule, R. S. Katti, and R. G. Kavasseri, "A variable length fast message authentication code for secure communication in smart grids," in *Proc. IEEE Power & Energy Society General Meeting*, 2012, pp. 1-6.
[18] D. Li, Z. Aung, J. R. Williams, and A. Sanchez, "Efficient authentication scheme for data aggregation in smart grid with fault tolerance and fault diagnosis," in *Proc. IEEE PES ISGT*, 2012, pp. 1-8.
[19] H. Nicanfar and V. C. M. Leung, "Multilayer consensus ECC-based password authentication key-exchange protocol for smart grid system," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 253-264, Mar. 2013.
[20] H. Nicanfar, P. Jokar, and V. Leung, "Efficient authentication and key management mechanisms for smart grid communications," *IEEE Systems Journal*, vol. 8, no. 2, pp. 629-640, Jun. 2014.
[21] S. Oh and J. Kwak, "Mutual authentication and key establishment mechanism using DCU certificate in smart grid," *Applied Mathematics & Information Sc.*, vol. 6, no. 1, pp. 257S-264S, 2012.
[22] H. Li, R. Lu, L. Zhou, B. Yang, and X. Shen, "An efficient merkle-tree-based authentication scheme for smart grid," *IEEE Systems Journal*, vol. 8, no. 2, pp. 655-662, 2014.
[23] A. R. Metke and R. L. Ekl, "Smart grid security technology," in *Proc. Innovative Smart Grid Technologies (ISGT)*, 2010, pp. 1-7.
[24] K. Kursawe, G. Danezis, and M. Kohlweiss, "Privacy-friendly aggregation for the smart grid," in *Proc. PET*, 2011, pp. 175-191.
[25] C. Bekara, T. Luckenbach, and K. Bekara, "A privacy preserving and secure authentication protocol for advanced metering infrastructure with non-repudiation service," in *Proc. ENERGY*, 2012, pp. 60-68.
[26] Y. Yan, R. Hu, and S. Das, "An efficient security protocol for advanced metering infrastructure in smart grid," *IEEE Network*, pp. 64-71, 2013.
[27] F. Luo, Z. Y. Dong, Y. Chen, Y. Xu, and K. P. Wong, "Hybrid cloud computing platform: the next generation IT backbone for smart grid," in *Proc. Power & Energy Society General Meeting*, 2012, pp. 1-7.
[28] S. Rusitschka, K. Eger, and C. Gerdes, "Smart grid data cloud: a model for utilizing cloud computing in the smart grid domain," in *Proc. Internl Conf. on Smart Grid Communications*, 2010, pp. 483-488.
[29] J. Baek, Q. H. Vu, J. K. Liu, X. Huang, and Y. Xiang, "A secure cloud computing based framework for big data information management of smart grid," *IEEE Trans. on Cloud Computing*, vol. 3, no. 2, pp. 233-244, Jun. 2015.
[30] S. Bera, S. Misra, and J. J. P. C. Rodrigues, "Cloud computing applications for smart grid: a survey," *IEEE Transactions on Parallel & Distributed Systems*, vol. 26, no. 5, pp. 1477-1494, May 2015.
[31] A. H. Mohsenian-Rad and A. Leon-Garcia, "Coordination of cloud computing and smart power grids," in *Proc. IEEE International Conference on Smart Grid Communications*, 2010, pp. 368-372.
[32] Y. Jiang, X. Guo, C. Li,, H. Wen, C. Lei, and Z. Rui, "An efficient and secure search database scheme for cloud computing in smart grid," in *Proc. Conf. on Communications & N/w Security*, 2013, pp. 413-414.
[33] B. Bitzer and E. S. Gebretsadik, "Cloud computing framework for smart grid applications," in *Proc. 48th International Universities' Power Engineering Conference (UPEC)*, 2013, pp. 1-5.
[34] Cyber Security Challenges in Using Cloud Computing in the Electric Utility Industry, Pacific Northwest National Laboratory, Prepared for the U.S. Department of Energy under Contract DE-AC05-76RL01830, PNNL-21724, Sept. 2012. [Online]. http://www.pnnl.gov/main/publications/external/ technical_reports/pnnl-21724.pdf.
[35] H. Li, Y. Dai, and H. Yang, "Identity-based authentication for cloud computing," in *Proc. CloudCom*, LNCS 5931, 2009, pp. 157-166.
[36] H. Lim and K. Paterson, "Identity-based cryptography for grid security," *International Journal of Information Security*, vol. 10, pp. 15-32, 2011.
[37] J. H. Oh, K. K. Lee, and S. Moon, "How to solve key escrow and identity revocation in identity-based encryption schemes," in *Proc. Information Systems Security*, LNCS 3803, 2005, pp. 290-303.
[38] J. Baek, R. Safavi-Naini, and W. Susilo, "Certificateless public key encryption without pairing," in *Proc. Info. Security*, 2005, pp. 134-148.
[39] M. Scott, "On the efficient implementation of pairing-based protocols," *IACR Cryptology*, pp. 334-346, 2011.
[40] D. R. L. Brown, SEC 1: Elliptic Curve Cryptography, Standards for Efficient Cryptography. [Online[. http://www.secg.org/sec1-v2.pdf.
[41] Smart Grid System Report, U.S. Department of Energy, July 2009. [Online]. smartgrid.gov/sites/default/files/resources/systems_report.pdf.
[42] N. Saxena and N. S. Chaudhari, "EasySMS: a protocol for end-to-end secure transmission of SMS," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 7, pp. 1157-1168, Jul. 2014.
[43] Security Guidelines for Cryptographic Algorithms in the W3C Web Cryptography API, W3C/MIT Cryptosense/INRIA, Nov. 2015. [Online]. https://www.w3.org/2012/webcrypto/draft-irtf-cfrg-webcrypto-algorithms-00#ECDSA.
[44] R. P. Gallant, R. J. Lambert, and S. A. Vanstone, "Faster point multiplication on elliptic curves with efficient endomorphisms," *IACR*, 2001. [Online]. https://www.iacr.org/archive/crypto2001/21390189.pdf.
[45] Measuring mobile broadband performance in the UK, 13 Nov. 2014. [Online]. http://stakeholders.ofcom.org.uk/binaries/research/broadband-research/mbb-nov14.pdf.
[46] J. Wessels, Applications of BAN-logic, CMG Finance, Apr. 2001. [Online]. win.tue.nl/ipa/archive/springdays2001/banwessels.pdf.

Netesh Saxena (S'10-M'14) is working as a Post-Doctoral Researcher at Georgia Institute of Technology, USA. Prior to this, he was with the State University of New York Korea as a Post-Doctoral Researcher and a Visiting Scholar at Stony Brook University, USA. He earned his Ph.D. in Computer Science & Engineering from the IIT Indore, India. In 2013-14, he was a Visiting Research Student and a DAAD Scholar at B-IT, Rheinische-Friedrich-Wilhelms Universitt, Bonn, Germany. He was also a TCS Research Scholar during Jan. 2012 - Apr. 2014. He works in the area of security and privacy. His current research interests include smart grid security, vehicle-to-grid security and privacy, cryptography, security and privacy in the cellular networks, and secure mobile applications. He has published several papers in various international peer-reviewed journals and conferences. He is a member of IEEE, ACM, and CSI.



Bong Jun Choi (S'09-M'11) received his B.Sc. and M.Sc. degrees from Yonsei University, Korea, both in electrical and electronics engineering, and the Ph.D. degree from University of Waterloo, Canada, in electrical and computer engineering. He is currently an assistant professor at the Department of Computer Science, State University of New York Korea, Korea, and jointly a research assistant professor at the Department of Computer Science, Stony Brook University, USA. His current research focuses on energy efficient networks, distributed mobile wireless networks, smart grid communications, and network security. He serves as an editor of KSII Transactions on Internet and Information Systems and a member of the Smart Grid Core Security Technology Development Steering Committee, Korea. He also serves on the technical program committees for many international conferences such as IEEE PECON, IFIP NTMS, and IEEE CMC. He is a member of the IEEE and the ACM.